

An upper bound on quantum fault tolerant thresholds

Jesse Fern

*Berkeley Center for Quantum Information and Computation,
Department of Mathematics, University of California, Berkeley, California, 94720 **

In this paper we calculate upper bounds on fault tolerance, without restrictions on the overhead involved. Optimally adaptive recovery operators are used, and the Shannon entropy is used to estimate the thresholds. By allowing for unrealistically high levels of overhead, we find a quantum fault tolerant threshold of 6.88% for the depolarizing noise used by Knill [16], which compares well to "above 3%" evidenced by Knill. We conjecture that the optimal threshold is 6.90%, based upon the hashing rate. We also perform threshold calculations for types of noise other than that discussed by Knill.

I. INTRODUCTION

In this paper, we calculate the quantum fault tolerant thresholds for the $[[7, 1, 3]]$ and $[[23, 1, 7]]$ CSS codes under massive amounts of post-selection. Post-selection [14] is the process of performing error detection, and then rejecting the result if an error is detected. We optimize the recovery operators under concatenation [7] to find the optimal thresholds. Here there are no restrictions on the quantity of overhead involved. Optimizations of the error threshold with respect to the overhead will be considered in future work.

Suppose that, given a logical data qubit $|\psi_D\rangle$ encoded into some physical data qubits and a logical ancilla qubit $|\psi_A\rangle$ encoded into some physical ancilla qubits, we wish to detect bit flip errors in the data qubits. We apply CNOT gates transversally from the data qubits to the ancilla qubits, and then measure the ancilla qubits, as shown in Fig. 1. The fact that the syndrome measurement may be faulty is included in our analysis. For post-selection, we start all over if an error is detected.

This process works well for CSS codes and bit flip qubits. CNOT is encoded as itself acting transversally on each pair of qubits. In terms of the logical qubits, the result will resemble Fig. 2. For an arbitrary logical data

qubit $|\overline{\psi_D}\rangle$, we use $|\overline{\psi_A}\rangle = |\overline{+}\rangle$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ for the logical ancilla qubit. If the logical source qubit is an encoded Pauli eigenstate of \overline{Z} , that is $|\overline{0}\rangle$ or $|\overline{1}\rangle$, then it is more efficient for the logical destination ancilla qubit to also be an encoded Pauli eigenstate of \overline{Z} .

We assume the ideal case that the noise on each qubit is independent, and that any bit flip error that was sent to or was already existing on the destination (ancilla) qubits can be detected. We model this by looking at each physical data qubit and physical ancilla qubit pair separately. However, it is not exactly true that all of these errors can be detected; if w is the lowest weight (that is, the number of qubits for which the Pauli matrix is not the identity on that qubit) of a stabilizer element (other than the identity) for the code, then there are some cases where a total of w errors in the source or destination qubits can result in an undetected stabilizer error in the destination qubits.

We run error detection separately for bit flip errors (X or Y) and phase flip errors (Z or Y). One way to do this is to apply Hadamard gates in between the error detection steps to change bit flip errors into phase flip errors, and vice versa. In this paper, we apply post-selection to reduce the rate of errors in some ancilla qubits.

In Sec. II, the noise resulting from a CNOT gate followed by measurement of the destination qubit is found. The same results are derived using the channel maps [7, 8] of quantum codes in Appendix A. Both of these methods assume the same noise on each qubit.

In Sec. III, we apply these results to a code encoded in some specific ancilla qubits that allow for an encoded CNOT with error correction using just one CNOT per logical data qubit. We again assume the same noise on each qubit.

In Sec. IV, we consider realistic types of noise, and calculate the relevant quantum fault tolerant thresholds.

FIG. 1: General bit flip error detection process. If we are performing post-selection, we start all over if an error is detected.

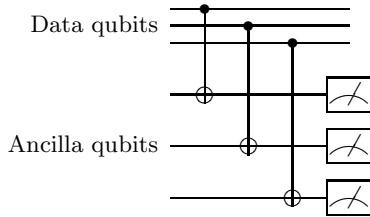
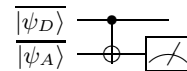


FIG. 2: Logical qubits under bit flip error detection



*Electronic address: jesse@math.berkeley.edu

In Sec. V, we examine how the fact that the noise is not the same on each qubit affects the threshold. We use the Shannon information entropy [21] as in [7] to calculate the actual thresholds.

In Sec. VI, we look at how well the noise is corrected when we are below the fault tolerant threshold.

II. BASIC CALCULATIONS

A. Channel Notation

A qubit has two orthogonal states, $|0\rangle$ and $|1\rangle$. If there is a set of states $|\psi_i\rangle$ with corresponding probabilities of occurring p_i , the resulting mixed state is represented by the density matrix

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|.$$

We can write this 2 by 2 matrix in terms of the 4 one qubit Pauli matrices I, X, Y and Z as

$$\rho = \frac{1}{2}(I + c_X X + c_Y Y + c_Z Z).$$

Alternately, one can write this density matrix in vector form as

$$v(\rho) = \begin{bmatrix} 1 \\ c_X \\ c_Y \\ c_Z \end{bmatrix}, \quad (1)$$

which we shall call a density vector. Now we consider the map from density matrices to density matrices. These maps are of the form

$$\rho \rightarrow \sum_j A_j \rho A_j^\dagger, \quad \text{where} \quad \sum_j A_j^\dagger A_j = I.$$

They can be written as the channel superoperator matrix that acts on the density vectors as

$$\mathcal{N}^{(1)} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ N_{XI} & N_{XX} & N_{XY} & N_{XZ} \\ N_{YI} & N_{YX} & N_{YY} & N_{YZ} \\ N_{ZI} & N_{ZX} & N_{ZY} & N_{ZZ} \end{bmatrix},$$

where the $N_{\sigma\sigma'}$ are all real.

A unitary matrix U acts upon a density state as $\rho \rightarrow U\rho U^\dagger$. We represent it in the channel superoperator form as a superoperator $\mathcal{O}(U)$. We represent these in the Pauli basis. A Pauli matrix σ has the diagonal superoperator $\mathcal{O}(\sigma)$, where each of the coefficients $\mathcal{O}(\sigma)_{\sigma'\sigma'}$ is 1 if σ and σ' commute, and -1 if they anti-commute. For the 1 qubit Pauli matrices, the superoperators are diagonal and are written as

$$\begin{aligned} \mathcal{O}(I) &= [1, 1, 1, 1] & \mathcal{O}(X) &= [1, 1, -1, -1] \\ \mathcal{O}(Y) &= [1, -1, 1, -1] & \mathcal{O}(Z) &= [1, -1, -1, 1] \end{aligned} \quad (2)$$

where we record the diagonal elements in vector form. We can write a diagonal channel in terms of the probabilities p_σ of having the Pauli error σ as

$$[1, N_{XX}, N_{YY}, N_{ZZ}] = \sum_\sigma p_\sigma \mathcal{O}(\sigma) = [1, x, y, z],$$

which, using Eq. 2, gives

$$\begin{aligned} x &= 1 - 2(p_Y + p_Z) \\ y &= 1 - 2(p_X + p_Z) \\ z &= 1 - 2(p_X + p_Y). \end{aligned} \quad (3)$$

B. Measurement channels

Proposition 1 *If we have a pure state $\nu = |\psi\rangle \langle \psi|$ and a state ρ , and we measure the state ρ in an orthonormal basis which includes $|\psi\rangle$, then the probability of measuring $|\psi\rangle$ is the fidelity, which is half of the inner product of the vectorized density matrices (Eq. 1):*

$$F(\rho, \nu) = \frac{1}{2} \langle v(\rho) | v(\nu) \rangle.$$

Proof: ρ can be written as a linear combination of pure states as $\rho = \sum_i d_i |\gamma_i\rangle \langle \gamma_i|$, and so the probability of measuring $|\psi\rangle$ is

$$F = \sum_i d_i \langle \gamma_i | \psi \rangle^2 = \sum_i d_i \langle \psi | \gamma_i \rangle \langle \gamma_i | \psi \rangle = \langle \psi | \rho | \psi \rangle.$$

Now, $\text{tr } \rho \nu = \sum_i \langle a_i | \rho \nu | a_i \rangle$ for any basis $|a_i\rangle$, so

$$\text{tr } \rho \nu = \text{tr } \rho | \psi \rangle \langle \psi | = \langle \psi | \rho | \psi \rangle \langle \psi | \psi \rangle = \langle \psi | \rho | \psi \rangle = F.$$

If

$$v(\rho) = \begin{bmatrix} 1 \\ \rho_X \\ \rho_Y \\ \rho_Z \end{bmatrix}, \quad v(\nu) = \begin{bmatrix} 1 \\ \nu_X \\ \nu_Y \\ \nu_Z \end{bmatrix},$$

then

$$\text{tr } \rho \nu = \frac{1}{2} \sum_\sigma \rho_\sigma \nu_\sigma = \frac{1}{2} \langle v(\rho) | v(\nu) \rangle.$$

■

From this it follows that:

Corollary 2 *If a noise \mathcal{N} is applied to the state ρ , the probability of measuring the pure state ν is*

$$\frac{1}{2} \langle v(\nu) | \mathcal{N} | v(\rho) \rangle.$$

For example, if they are both the pure state $|0\rangle$, $\rho = \nu = \frac{1}{2}(I + Z)$, and so

$$\frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \mathcal{N} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{N_{II} + N_{IZ} + N_{ZI} + N_{ZZ}}{2}.$$

TABLE I: Measurement probabilities

Measured state	Initial state $\lambda_i = 1$	Initial state $\lambda_i = -1$
$\lambda_m = 1$	$\frac{N_{II} + N_{I\sigma} + N_{\sigma I} + N_{\sigma\sigma}}{2}$	$\frac{N_{II} - N_{I\sigma} + N_{\sigma I} - N_{\sigma\sigma}}{2}$
$\lambda_m = -1$	$\frac{N_{II} + N_{I\sigma} - N_{\sigma I} - N_{\sigma\sigma}}{2}$	$\frac{N_{II} - N_{I\sigma} - N_{\sigma I} + N_{\sigma\sigma}}{2}$

Theorem 3 *If a noise \mathcal{N} is applied to a qubit, and then the qubit is measured in the Pauli operator σ basis, which yields one of the Pauli eigenstates $|\sigma_+\rangle$ or $|\sigma_-\rangle$, the correct result is obtained by the measurement with probability*

$$m_c = \frac{1 + N_{\sigma\sigma}}{2}.$$

Proof: We have initial state ρ_i and measured state ρ_m , which are eigenstates of σ , with associated eigenvalues λ_i and λ_m . Then, from Cor. 2, after a noise \mathcal{N} occurs, the probability of measuring the initial state ρ_i as the measured state ρ_m is

$$\begin{aligned} \frac{1}{2} \langle v(\rho_m) | \mathcal{N} | v(\rho_i) \rangle &= \frac{1}{2} [1 \quad \lambda_m] \begin{bmatrix} N_{II} & N_{I\sigma} \\ N_{\sigma I} & N_{\sigma\sigma} \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_i \end{bmatrix} \\ &= \frac{N_{II} + \lambda_1 N_{I\sigma} + \lambda_2 N_{\sigma I} + \lambda_1 \lambda_2 N_{\sigma\sigma}}{2}. \end{aligned}$$

Tab. I gives the probabilities for the 4 cases in which the initial eigenvalue λ_i is either -1 or $+1$ and the measured eigenvalue λ_m is either -1 or $+1$.

The total probability of a wrong measurement averages to $\frac{1 - N_{\sigma\sigma}}{2}$, and the total probability of a correct measurement averages to $\frac{1 + N_{\sigma\sigma}}{2}$. ■

C. Noise from applying CNOT

As shown in Fig. 3, we have a source qubit $|\psi\rangle_S$ with pre-existing local noise \mathcal{S} , and a destination qubit $|\psi\rangle_D$ with pre-existing local noise \mathcal{D} . When we apply CNOT, the two qubit noise \mathcal{Q} occurs on these qubits. The total superoperator for this process is found by multiplying the initial noise $\mathcal{S} \otimes \mathcal{D}$ first by the superoperator for CNOT and then by the new noise \mathcal{Q} from CNOT. This results in

$$\mathcal{T} = \mathcal{Q} \circ \mathcal{O}(\text{CNOT}) \circ (\mathcal{S} \otimes \mathcal{D}),$$

where \circ represents the matrix multiplication of superoperators. Since we wanted only CNOT to be applied, the total noise can be represented in terms of a noise superoperator \mathcal{N} as

$$\mathcal{T} = \mathcal{N} \circ \mathcal{O}(\text{CNOT}) = \mathcal{Q} \circ \mathcal{O}(\text{CNOT}) \circ (\mathcal{S} \otimes \mathcal{D}) \circ \mathcal{O}(\text{CNOT}). \quad (4)$$

D. Diagonal noise

In the rest of this paper, we assume that the noise is diagonal. We utilize the shorthand $\mathcal{N}_\sigma \equiv \mathcal{N}_{\sigma\sigma}$. This noise

TABLE II: The noise $\mathcal{R} = \mathcal{O}(\text{CNOT}) \circ (\mathcal{S} \otimes \mathcal{D}) \circ \mathcal{O}(\text{CNOT})$ resulting from $\mathcal{S} \otimes \mathcal{D}$ conjugated with CNOT.

σ	$\mathcal{R}_{\sigma I}$	$\mathcal{R}_{\sigma X}$	$\mathcal{R}_{\sigma Y}$	$\mathcal{R}_{\sigma Z}$
I	$S_I D_I$	$S_I D_X$	$S_Z D_Y$	$S_Z D_Z$
X	$S_X D_X$	$S_X D_I$	$S_Y D_Z$	$S_Y D_Y$
Y	$S_Y D_X$	$S_Y D_I$	$S_X D_Z$	$S_X D_Y$
Z	$S_Z D_I$	$S_Z D_X$	$S_I D_Y$	$S_I D_Z$

TABLE III: The total noise \mathcal{N} resulting from the error detection process.

σ	$\mathcal{N}_{\sigma I}$	$\mathcal{N}_{\sigma X}$	$\mathcal{N}_{\sigma Y}$	$\mathcal{N}_{\sigma Z}$
I	$S_I D_I Q_{II}$	$S_I D_X Q_{IX}$	$S_Z D_Y Q_{IY}$	$S_Z D_Z Q_{IZ}$
X	$S_X D_X Q_{XI}$	$S_X D_I Q_{XX}$	$S_Y D_Z Q_{XY}$	$S_Y D_Y Q_{XZ}$
Y	$S_Y D_X Q_{YI}$	$S_Y D_I Q_{YX}$	$S_X D_Z Q_{YY}$	$S_X D_Y Q_{YZ}$
Z	$S_Z D_I Q_{ZI}$	$S_Z D_X Q_{ZX}$	$S_I D_Y Q_{ZY}$	$S_I D_Z Q_{ZZ}$

maps a density matrix $I + \sigma \rightarrow I + N_\sigma \sigma$. The noise on the source qubits has diagonal noise $\mathcal{S} = [1, S_X, S_Y, S_Z]$, and the noise on the destination qubits has diagonal noise $\mathcal{D} = [1, D_X, D_Y, D_Z]$.

If the off-diagonal terms are small, they can be neglected anyway. If they are $o(\epsilon)$, the threshold is only affected by $o(e^m)$, where m is the minimum weight of the non-identity elements that stabilize the code [8]. For the $[[7, 1, 3]]$ CSS code, $m = 4$. For the $[[23, 1, 7]]$ CSS code, $m = 8$.

To determine how the CNOT gate effects the errors, we look at the map of the Pauli matrices under conjugation by CNOT. This gives the map

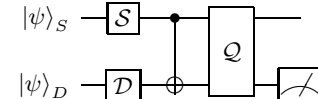
$$\begin{aligned} (II), (IX), IY &\leftrightarrow ZY, IZ \leftrightarrow ZZ, XI \leftrightarrow XX \\ XY &\leftrightarrow YZ, XZ \leftrightarrow YY, YI \leftrightarrow YX, (ZI), (ZX). \end{aligned}$$

The diagonal noise $\mathcal{R} = \mathcal{O}(\text{CNOT}) \circ (\mathcal{S} \otimes \mathcal{D}) \circ \mathcal{O}(\text{CNOT})$ is given by Table II. From this, the resulting diagonal noise $\mathcal{N} = \mathcal{Q} \circ \mathcal{R}$ is given in Table III.

1. After Z measurement on destination qubit

Now we assume that the destination qubit is set up so that we can detect a Z error upon measuring it in the Z

FIG. 3: A pair of physical qubits under error detection. Pre-existing noise on the source qubit is given by \mathcal{S} , pre-existing noise on the destination qubit by \mathcal{D} . The noise from the CNOT gate is \mathcal{Q} .



basis. From Theorem 3, we have two types of syndromes, the "correct" one which corresponds to no error detected, and the "incorrect" one which corresponds to an error detected. These have probabilities $\frac{1+Z}{2}$ and $\frac{1-Z}{2}$. By tracing out N on the destination qubit, these result in

$$\mathcal{G}_\sigma^{II} = \frac{N_{\sigma I} + N_{\sigma Z}}{2} \quad \mathcal{G}_\sigma^{IX} = \frac{N_{\sigma I} - N_{\sigma Z}}{2}. \quad (5)$$

This gives

$$\mathcal{G}^{II} = \frac{1}{2}(A + B) \quad \text{and} \quad \mathcal{G}^{IX} = \frac{1}{2}(A - B), \quad (6)$$

where

$$A = [1, S_X D_X Q_{XI}, S_Y D_X Q_{YI}, S_Z Q_{ZI}]$$

$$B = [S_Z D_Z Q_{IZ}, S_Y D_Y Q_{XZ}, S_X D_Y Q_{YZ}, D_Z Q_{ZZ}]$$

are the 1st and 4th columns of Tab. III, respectively.

2. After X measurement on source qubit and Z measurement on destination qubit

Like before, we trace out, this time with $\frac{1 \pm X}{2}$. Tracing out Eq. 6, we get probabilities of Pauli errors

$$\begin{bmatrix} p_I \\ p_X \\ p_Y \\ p_Z \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ S_X D_X Q_{XI} \\ S_Y D_Y Q_{XZ} \\ S_Z D_Z Q_{IZ} \end{bmatrix}. \quad (7)$$

Note that only bit flip errors on the destination qubit and phase flip errors on the source qubit contribute to these errors.

E. Measurement errors

Suppose that we have probability p_m of an X measurement error. This is equivalent to probability $p_X = p_m$ of X error and probability $p_I = 1 - p_m$ of an I error (no error) right before measurement. From Eq. 2, this has the channel

$$[1, x, y, z] = (1 - p_m)\mathcal{O}(I) + p_m\mathcal{O}(X) = [1, 1, m, m],$$

where

$$m = 1 - 2p_m. \quad (8)$$

Similarly, Z measurement errors produce the channel $[1, x, y, z] = [1, m, m, 1]$.

This changes the resulting noise after measurement from Eq. 6 to

$$\mathcal{G}^{II} = \frac{1}{2}(A + mB) \quad \text{and} \quad \mathcal{G}^{IX} = \frac{1}{2}(A - mB).$$

Then Eq. 7 becomes

$$\begin{bmatrix} p_I \\ p_X \\ p_Y \\ p_Z \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ mS_X D_X Q_{XI} \\ m^2 S_Y D_Y Q_{XZ} \\ mS_Z D_Z Q_{IZ} \end{bmatrix}.$$

III. SPECIAL ANCILLA QUBITS

Suppose that we have a high distance (minimum weight of an undetected error) code. We have found ancilla qubit encodings that allow for a higher quantum fault tolerant threshold than other methods [16]. These ancilla qubits are logical qubits encoded in a high distance quantum code. We then use quantum teleportation to bring in logical data qubits.

Once we have sufficiently reduced the rates of errors in these qubits via post-selection, we use quantum teleportation [1] to replace the logical ancilla qubits with logical data qubits, as is also done in the work of Knill [14] and Reichardt [19]. The simplest teleportation is to use a Bell pair (two qubit cat state)

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \sum_{a=0}^1 |aa\rangle,$$

and apply the standard teleportation gate given in Fig. 4. If $|b\rangle$ represents the standard basis elements of $|\psi\rangle$, then we start with $|b\rangle |a\rangle |a\rangle$. After CNOT is applied from the first to the second qubit, it becomes $|b\rangle |a \oplus b\rangle |a\rangle$. Measuring the second qubit results in $|b\rangle |b\rangle$ if 0 is measured, and $|b\rangle \oplus |b \oplus 1\rangle$ if 1 is measured. Applying X to the last qubit if 1 is measured results in $|b\rangle |b\rangle$. The last measurement and conditional gate gives either $|b\rangle$ or the initial $|\psi\rangle$ state. The three steps in Fig. 4 can be represented as

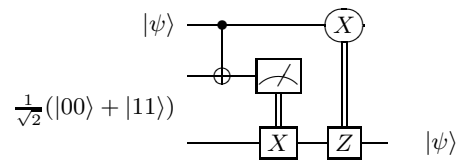
$$|b\rangle |a\rangle |a\rangle \rightarrow |b\rangle |a \oplus b\rangle |a\rangle \rightarrow |b\rangle |b\rangle \rightarrow |b\rangle.$$

While this both inputs and outputs $|\psi\rangle$, the advantage is that the output can be a post-selected logical qubit, which can be used to correct errors in the physical qubits that comprise an arbitrary logical data qubit.

Now, if we have the 3 qubit cat state

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) = \frac{1}{\sqrt{2}} \sum_{a=0}^1 |aaa\rangle,$$

FIG. 4: Teleportation circuit



and apply a similar process shown in Fig. 5, then after each of the 3 steps, we have

$$|b\rangle |a\rangle |a\rangle |a\rangle \rightarrow |b\rangle |a \oplus b\rangle |a\rangle |a\rangle \rightarrow |b\rangle |b\rangle |b\rangle \rightarrow |b\rangle |b\rangle.$$

The effect of this is to send $|0\rangle$ to $|00\rangle$ and $|1\rangle$ to $|11\rangle$ with freshly post-selected qubits.

In the Hadamard basis, the 3 qubit cat state is transformed to

$$\frac{1}{2}(|000\rangle + |011\rangle + |101\rangle + |110\rangle) = \frac{1}{2} \sum_{a=0}^1 \sum_{b=0}^1 |a\rangle |b\rangle |a \oplus b\rangle.$$

In Fig. 6, a circuit is shown that uses 2 teleportations to bring in two additional states with the basis elements $|c\rangle$ and $|d\rangle$. After each step in this circuit, this results in

$$\begin{aligned} |c\rangle |d\rangle |a\rangle |b\rangle |a \oplus b\rangle &\rightarrow |c\rangle |d\rangle |a \oplus c\rangle |b\rangle |a \oplus b\rangle \\ &\rightarrow |c\rangle |d\rangle |a \oplus c\rangle |b \oplus d\rangle |a \oplus b\rangle \rightarrow |c\rangle |d\rangle |b \oplus d\rangle |c \oplus b\rangle \\ &\rightarrow |c\rangle |d\rangle |c \oplus d\rangle \rightarrow |d\rangle |c \oplus d\rangle \rightarrow |c \oplus d\rangle. \end{aligned}$$

If these two previous ancillas are used together, they give

$$|a\rangle |b\rangle \rightarrow |a\rangle |a\rangle |b\rangle \rightarrow |a\rangle |a \oplus b\rangle,$$

which is CNOT. Alternately, we can combine the two ancillas via teleportation to get the 4 qubit state

$$|a\rangle |b\rangle |a\rangle |a \oplus b\rangle = \frac{1}{2}(|0000\rangle + |1011\rangle + |0101\rangle + |1110\rangle),$$

and teleport $|a\rangle$ and $|b\rangle$ into the first and second qubits respectively as illustrated in Fig. 7.

For CSS codes, we can assume that noise from a logical CNOT acting on two logical data qubits determines the fault tolerant thresholds. See section B for a discussion of the other gates necessary for universal quantum computation.

A. Repeated post-selection

This section discusses the process of repeated post-selection. By performing a large number of post-selections, the resulting noise decreases in magnitude, converging to some value.

The process which uses ancilla qubits can be broken down into two parts. First, we run post-selection repeatedly to refine ancilla qubits. This process works by detecting bit flip errors as in Fig. 1, then switching to the Hadamard basis and again applying the circuit of Fig. 1 to detect what were originally phase flip errors. As post-selection is repeatedly applied, the noise will converge to some fixed channel that can be estimated by iterative calculation. Second, we use teleportation to bring a logical data qubit into the ancilla qubits; simultaneously this applies a logical CNOT.

a. Noise under post-selection When we are running error detection with bit flip errors, we apply CNOT transversally from the source qubits to the destination qubits. Bit flip errors will propagate from the first to the second qubit, just like we want them to. However, phase flip errors will propagate backwards. To deal with this, we minimize the phase flip errors in the destination qubits. The source qubits will also have been post-selected for phase flip errors in the previous step. Therefore, before we apply the CNOT, we can assume that there is the same diagonal noise on the source and destination qubits. We let $\sigma = S_\sigma = D_\sigma$. Then using Eq. 6, the noise after CNOT followed by a post-selected measurement (measurement with no error) becomes

$$\begin{aligned} x_{\text{out}} &= \frac{x^2 Q_{XI} + y^2 Q_{XZ}}{1 + z^2 Q_{IZ}} \\ y_{\text{out}} &= xy \frac{Q_{YI} + Q_{YZ}}{1 + z^2 Q_{IZ}} \\ z_{\text{out}} &= z \frac{Q_{ZI} + Q_{ZZ}}{1 + z^2 Q_{IZ}}. \end{aligned}$$

In the next step, we run error detection on the other type of noise (bit flip or phase flip). This can be represented by applying a Hadamard gate to change bit flip errors into phase flip errors and vice versa, and then applying the same error detection process. This gives $x_{\text{out}} = z$, $z_{\text{out}} = x$, $y_{\text{out}} = y$, yielding the map

$$[x, y, z] \rightarrow [z_{\text{out}}, y_{\text{out}}, x_{\text{out}}]. \quad (9)$$

Since we just post-selected for bit flip errors, after the Hadamard there is a low rate of phase flip errors $p_f = p_X + p_Y$. From Eq. 3, we see that $x = 1 - 2p_f$ is close to 1.

FIG. 5: Teleportation splitting circuit.

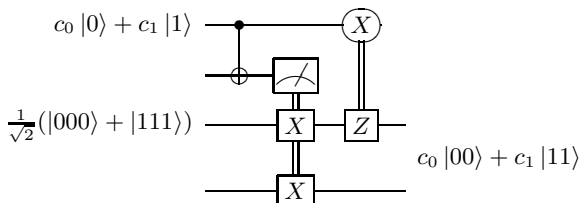
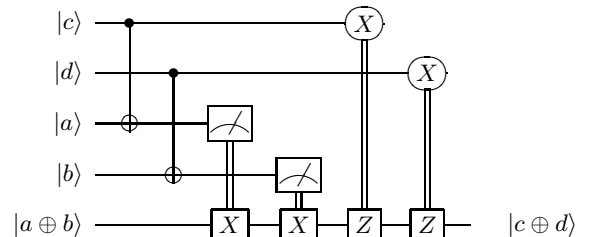


FIG. 6: Teleportation merging circuit.



Under repeated post-selection, the noise converges to a solution $[1, x, y, z]$ of

$$\begin{aligned} (1 + z^2 Q_{IZ})^2 &= z(Q_{YI} + Q_{YZ})(Q_{ZI} + Q_{ZZ}) \\ x &= \frac{1 + z^2 Q_{IZ}}{Q_{YI} + Q_{YZ}} \\ y^2 Q_{XZ} &= z(1 + z^2 Q_{IZ}) - x^2 Q_{XI}. \end{aligned}$$

This is the effective noise after infinity iterations of post-selection.

b. Teleportation Once there has been sufficient post-selection on the logical source qubit, we apply **CNOT** from it to a logical destination qubit, and then measure the physical destination qubits in the normal Z basis (which detects bit flip errors) and the source qubits in the X basis (which detects phase flip errors) as shown in Fig. 4. It is important that the two logical qubits which will have a **CNOT** applied between them are post-selected last for opposite types of errors, one phase flip, one bit flip. This results in bit flip errors and phase flip errors being equally probable — useful for CSS codes which are very inefficient at correcting noise of only one of those types at a time.

Therefore, $S_Y = D_Y$, $S_X = D_Z = x$ (or z) $S_Z = D_X = z$ (or x). Consider a pair of physical qubits that are about to undergo an X measurement and a Z measurement as part of the teleportation circuit shown in Fig. 4. A bit flip error (X or Y errors) in the destination qubit results in an X error. A phase flip error (Z or Y errors) in the source qubit results in a Z error. If both occur, we have a Y error; and if neither occur, then no error. Thus, the probabilities of Eq. 7 are

$$\begin{bmatrix} p_I \\ p_X \\ p_Y \\ p_Z \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ mxzQ_{XI} \\ m^2y^2Q_{XZ} \\ mxzQ_{IZ} \end{bmatrix}, \quad (10)$$

where m is defined in terms of the probability of a measurement error in Eq. 8.

Since we assume independent noise on each qubit, we can judge how well the quantum code corrects this noise by using the optimal recovery adaptive concatenation

TABLE IV: The probability of each of the 16 two qubit Pauli errors resulting from a **CNOT** gate for depolarizing noise and independent noise. Note that the probability of a measurement error p_m is a separate parameter. The depolarizing noise used in this paper has no measurement errors; Knill noise is depolarizing noise with a probability $p_m = \frac{4}{15}p$ of a measurement error. Forward type noise has no measurement errors.

Prob.	Depolarizing	Forward	Independent noise
p_{II}	$1 - p$	$(1 - p_f)^2$	$(1 - p_f)^2(1 - p_b)^2$
p_{IX}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f(1 - p_b)^2$
p_{IY}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f(1 - p_b)p_b$
p_{IZ}	$\frac{p}{15}$	$(1 - p_f)^2$	$(1 - p_f)^2(1 - p_b)p_b$
p_{XI}	$\frac{p}{15}$	$(1 - p_f)^2$	$(1 - p_f)^2(1 - p_b)p_b$
p_{XX}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f(1 - p_b)p_b$
p_{XY}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f p_b^2$
p_{XZ}	$\frac{p}{15}$	$(1 - p_f)^2$	$(1 - p_f)^2 p_b^2$
p_{YI}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f(1 - p_b)p_b$
p_{YX}	$\frac{p}{15}$	p_f^2	$p_f^2(1 - p_b)p_b$
p_{YY}	$\frac{p}{15}$	p_f^2	$p_f^2 p_b^2$
p_{YZ}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f p_b^2$
p_{ZI}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f(1 - p_b)^2$
p_{ZX}	$\frac{p}{15}$	p_f^2	$p_f^2(1 - p_b)^2$
p_{ZY}	$\frac{p}{15}$	p_f^2	$p_f^2(1 - p_b)p_b$
p_{ZZ}	$\frac{p}{15}$	$(1 - p_f)p_f$	$(1 - p_f)p_f(1 - p_b)p_b$

TABLE V: Noise that gives the hashing bound (Shannon entropy of 1) under repeated post-selection. The p_{in} is the input rate of errors for the types of noise as defined in this paper. The p_σ are the probabilities of a Pauli error σ on each qubit as given in Eq. 10. Refer to Tab. IV for descriptions of the types of noise. These values were determined by numerical calculations.

	Depolarizing	Knill	Forward
p_{in}	8.27515%	6.90240%	4.81816%
$p_X = p_Z$	7.13361%	7.52699%	9.79217%
p_Y	4.78136%	4.12990%	1.21061%

technique of [7]. These are calculated in the next section.

IV. THRESHOLD CALCULATIONS FOR UNCORRELATED NOISE ON EACH QUBIT

The probabilities $p_{\sigma\sigma'}$ of a Pauli error $\sigma\sigma'$ for the types of noise resulting from **CNOT** that are discussed in this paper are given in Tab. IV.

A. Depolarizing noise

Suppose that the **CNOT** gate has probability $\frac{p}{15}$ of each of the non-identity two qubit Pauli errors. The total

FIG. 7: **CNOT** gate via Teleportation. $|\psi_3\rangle$ is $|\psi_1\rangle$ **CNOT** $|\psi_2\rangle$.

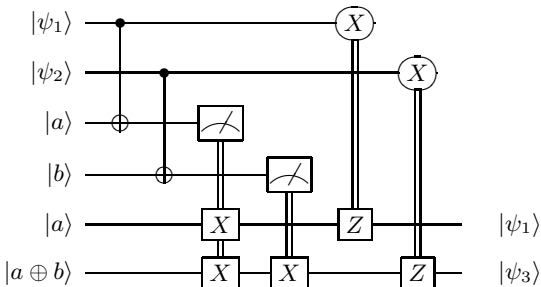


TABLE VI: Fault tolerant thresholds in terms of p_{in} assuming independent noise on each qubit. Refer to Tab. IV for descriptions of the types of noise.

Code	Depolarizing	Knill	Forward
Hashing	8.2751%	6.9024%	4.8182%
[[7, 1, 3]]	8.229(7)%	6.864(5)%	4.8036%
[[17, 1, 5]]	8.2(0)%	6.8%	4.790%
[[23, 1, 7]]	8.25%	6.88%	4.805%

TABLE VII: Capacity of codes for (p_X, p_Y, p_Z) type noise

Code	$(0, 0, p)$	(p, p, p)
Hashing	11.0028%	6.3097%
[[7, 1, 3]]	10.963(2)%	6.270(4)%
[[17, 1, 5]]	10.927(0)%	6.251 %
[[23, 1, 7]]	10.968%	6.29%
[[4, 2, 2]] and [[6, 2, 2]]	10.9466%	6.271(9)%

probability of no error is therefore $1 - p$. This results in $Q_{\sigma\sigma'} = 1 - \frac{16}{15}p$, except for $Q_{II} = 1$.

We look at two possibilities for measurement errors. We refer to the case where there are no measurement errors as depolarizing noise. In this case, $p_m = 0$ and $m = 1$. Knill noise [15, 16] refers to the case where there is probability $p_m = \frac{4}{15}p$ of a measurement error, which results in $m = 1 - \frac{8}{15}p$.

We first assume the same independent noise on each qubit, and calculate the resulting such noise from Eq. 10. We find the noise corresponding to the Hashing bound for these noises in Tab. V, that is, when the Shannon entropy of the noise is 1. We then determine whether a given code can correct this noise under repeated concatenation with itself by using the optimal recovery operator method of [7], which considers syndrome information from previous levels to determine the optimal recovery operator at each level, and uses Monte Carlo simulation to estimate the threshold. The stabilizers for the codes studied in Tab. V are given in [11].

In Fig. 8, we plot the effect of the measurement errors $p_m = \frac{4}{15}rp$ (that is, the fraction r of the Knill measurement error) on the fault tolerant threshold (assuming the hashing bound).

B. Independent bit flip and phase flip noise

In this section, we consider noise from the CNOT gate, where bit flips and phase flips act independently of each other, and the resulting noise on the source and destination qubits are also independent of each other. Note that since bit flip and phase flip errors are independent, their error detection and correction are done independently of each other. The calculations in this section can be per-

formed directly from Eqs. 9 and 10; however, because they are simpler in the case discussed here, we will go into more detail.

The probability of a phase error on the source qubit and the probability of a bit flip error on the destination qubit are both designated p_f . The probability of a bit flip error on the source qubit and the probability of a phase flip error on the destination qubit are both designated p_b . The probability of a measurement error is p_m . The probabilities of each of the Pauli errors for this type of noise are calculated in Table IV. For example, since there is probability $1 - p_f$ of no bit flip error on the destination qubit, probability $1 - p_f$ of no phase flip error on the source qubit, probability $1 - p_b$ of no phase error on the destination qubit, and probability p_b of a bit flip error on the source qubit, it follows that $p_{XI} = (1 - p_f)^2(1 - p_b)p_b$.

Let p_g be the probability of a bit flip or phase flip error immediately after post-selection is performed for that type of error. Let p_b be the probability of an error before that post-selection is run, i.e. after the other type of error detection is run.

We use the noise notation from Appendix C. Before we run post-selection, we have the noise $x_b = 1 - 2p_b$ on both the source and destination qubits. After we apply CNOT, the destination qubit gains the noise $f = 1 - 2p_f$ representing the probability of forward type error from CNOT. Measuring it results in the noise $m = 1 - 2p_m$, representing the probability of a measurement error. Therefore the total destination qubit noise is $x_g fm$. After post-selection, we must add the backwards type noise from CNOT to the source qubit. From Sec. C, we have

$$x_g = b \text{ post}(x_b, x_b fm) = b \frac{x_b + x_b fm}{1 + x_b^2 fm}.$$

When we run post-selection on the other type of noise (bit flip or phase flip), the x_g noise of this type back-actions from the ancilla qubits. In addition, the forward type noise contributes f noise, and so the total noise after the other type of error detection becomes

$$x_b = x_g^2 f = b^2 f x_b^2 \left(\frac{1 + fm}{1 + x_b^2 fm} \right)^2$$

before we run error detection on this type of noise (either bit flip or phase flip) again.

After repeated error detection and post-selection, the noise will tend towards an equilibrium value, which we calculate iteratively.

After the logical qubits have been sufficiently refined, we start the teleportation process given in Fig. 7. Applying CNOT results in a back-action. To improve the threshold, we have the source and data qubits optimized for opposite types of errors, so the back-action together with the regular noise results in $x_g x_b$ noise. In addition, the CNOT gate gives f noise, and the actual measurement gives m noise, resulting in a total combined noise from teleportation of

$$x = x_b x_g f m = x_g^3 f^2 m. \quad (11)$$

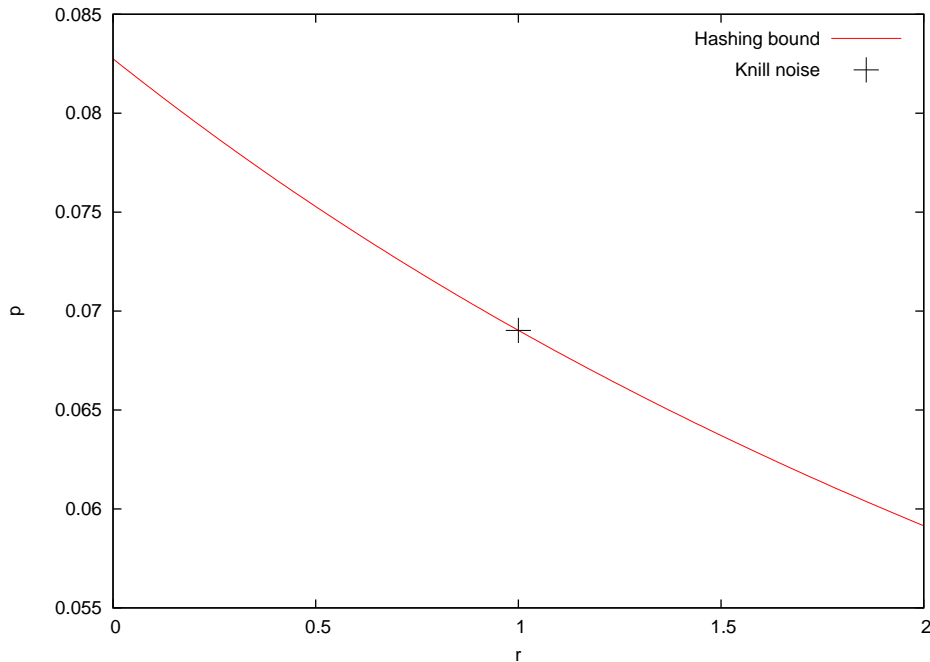


FIG. 8: The hashing bound for fault tolerant depolarizing noise. The CNOT gate has probability $\frac{p}{15}$ of each non-identity two qubit Pauli error occurring. The probability of a measurement error is $p_m = \frac{4}{15}rp$, where the $r = 1$ gives the measurement error used by Knill [16].

For CSS codes, the measurement process detects both X and Z errors, and is equally good at correcting both. If there are independent probabilities p of a bit flip and a phase flip error, then $(p_X, p_Y, p_Z) = (p - p^2, p^2, p - p^2)$. We consider for which types of noise the code can correct, assuming no additional errors during error detection and correction, as in [7]. For CSS codes, this has the same threshold as noise in the form $(p_X, p_Y, p_Z) = (0, 0, p)$. In Tab. VII, we calculate the thresholds for various CSS codes; these are found to be slightly below the hashing bound of $p = 11.0028\%$ found in [7].

We define forward type noise to consist of probability p_f of forward noise, with back noise probability p_b and measurement noise probability p_m both 0. Therefore, $n = m = 1$ and $f = 1 - 2p_f$. At the hashing bound, the noises are $f = 0.90363682$ (probability $p_f = 4.818159\%$ of a forward type noise), $x_g = 0.98482389$ and $x_b = 0.87641757$. The combined hashing bound noise is $c = x_g x_b f = x_g^3 f^2 = 0.77994427$, that is $p_c = \frac{1-c}{2}$ is the hashing bound.

V. ENTROPY-BASED CORRELATED NOISE CALCULATIONS

In the previous parts of this paper, we assumed that there is the same local uncorrelated noise on each qubit. However, degeneracies in the code caused by low weight stabilizer elements will introduce correlated errors that are noisier than the noise predicted by the uncorrelated

noise model. The most significant degeneracies occur at the first level of the code. In this section, we calculate the actual noise after post-selection at the first level (and sometimes look at the weaker effects at the second level) of the code. We use the entropy of this resulting correlated noise to evaluate the fault tolerant threshold. We then compare these exact results to those of the uncorrelated case. We see how the differences result from the degeneracies that arise from the low weight stabilizer elements.

A. Entropy of the $[[7, 1, 3]]$ code with forward type noise

The $[[7, 1, 3]]$ code is discussed in Appendix D.

Since the bit flip and phase flip errors are corrected independently of each other for this type of noise, we only have to consider correction of the one type of error. There can either be no error detected, in which case there was either a distance 0 or a distance 3 error, or there are 7 equivalent syndromes corresponding to the detection of an error, in which case a distance 1 or a distance 2 error occurred. If p_n are the probabilities of distance n errors,

then the Shannon entropy is

$$\begin{aligned} & (p_0 + p_3)(f(\frac{p_0}{p_0 + p_3}) + f(\frac{p_3}{p_0 + p_3})) \\ & + (p_1 + p_2)(f(\frac{p_1}{p_1 + p_2}) + f(\frac{p_2}{p_1 + p_2})) \\ & = \sum_n f(p_n) - f(p_0 + p_3) - f(p_1 + p_2), \end{aligned}$$

where

$$f(x) = -x \log_2 x.$$

We use the results of Sec. D to perform calculations in terms of the syndromes of the $[[7, 1, 3]]$ code that are similar to that of Sec. IV B.

Tab. VII shows that the $[[7, 1, 3]]$ CSS code can correct $(0, 0, p)$ type noise for $p < 10.963(2)\%$. This noise has an entropy of $e_1 = \frac{1}{2}1.0020(5)$ after the first level, and $e_2 = \frac{1}{2}1.0011(8)$ after the 2nd level.

We use the processes described in this section to calculate what happens to one level of the $[[7, 1, 3]]$ code in terms of the probabilities of a distance d error, instead of assuming the same noise on each qubit. We calculate for which value of p_f the entropy after the second level of the code is the same e_2 as before, in the uncorrelated case. This gives a threshold for forward type noise of $p_f = 4.6700\%$. As a check, we note that the entropy after the 1st level was $\frac{1}{2}1.0020(8)$, almost the same as before. By repeatedly concatenating the resulting noise for several levels, calculations have confirmed that this is indeed the new threshold.

B. Analysis of the difference

In the previous section, we calculate that the actual threshold for forward type noise for the $[[7, 1, 3]]$ code is $p_f = 4.6700\%$. This compares to the original approximation of $p_f = 4.8036\%$ found in Tab. VI.

The difference occurs because the code is slightly degenerate; there exist cases where the error that would be measured in the destination ancilla qubit corresponds to a stabilizer element, and therefore is not detected as an error. From Tab. XIII, it can be seen that the $[[7, 1, 3]]$ code has 7 stabilizer elements of weight 4. Thus, if there are 2 errors in the source qubits and 2 errors in the destination qubits, there can be an undetected weight 2 error in the source qubits. For each stabilizer element, this has an approximate probability $\binom{4}{2}p_g^2$ of occurring, where p_g is the probability of an undetected error from Sec. IV B. Together these give a factor of $42p_g^2$ difference.

In Eq. 11, we see that errors left undetected (and uncorrected) after post-selection contribute 3 times to the combined rate of noise analyzed for correctability by the code. Therefore, the total estimated difference in the probability of an encoded error (crash probability) is $c_e = 126p_g^2$.

TABLE VIII: Analysis of difference for $[[7, 1, 3]]$ code. p_a are the actual thresholds found from analyzing the correlated noise resulting from post-selection at the first level of the code. (However, later we find that the threshold for the forward noise is $p_a = 4.6699\%$.) Refer to Tab. IV for descriptions of the types of noise.

Value	Depolarizing	Knill	Forward
p_e	8.229(7)%	6.864(5)%	4.8036%
p_a	8.1096%	6.7785%	4.6700%
e_2	1.0012(1)	1.0011(9)	1.0011(7)
p_s	5.58%	4.69%	5.93%
p_d	9.42%	9.48%	10.04%
p_g	0.61%	0.51%	0.70%
c_e	0.47%	0.33%	0.62%
c_a	0.39%	0.32%	0.72%

If $p_f = 4.6700\%$, then $p_g = 0.70\%$, and so the estimated difference is $c_e = 0.62\%$. To calculate the actual difference, we understand that if there is the same noise x on each qubit, the resulting noise map for an encoded crash is $f_7(x) = \frac{7}{4}x^3 - \frac{3}{4}x^7$ [8]. For $p_f = 4.6700\%$, $x = 0.78794(5)$, and $f_7(x) = 0.7147$. For $p_f = 4.8036\%$, $x = 0.78073(6)$, and $f_7(x) = 0.7002$. The difference in crash probabilities is half the difference of $f_7(x)$, and is $c_a = 0.72\%$.

C. Further calculations

In Tab. VIII, we perform similar calculations for depolarizing and Knill noise. p_e is the threshold estimated by assuming the same noise on each qubit. e_2 is the logical entropy at the 2nd level of the code using this noise. p_a is the actual threshold found by fully post-selecting at the first level of the $[[7, 1, 3]]$ CSS code. Using the same noise on each qubit with the actual probability of an error p_a : p_s and p_d are the probabilities of source and destination qubit errors immediately before post-selection. p_g is the probability of an undetected error after post-selection. $c_e = 126p_g^2$ is the estimated difference in crash probabilities for a bit flip error (X or Y). Phase flip errors (Y or Z) have the same probabilities. c_a is the actual (not the estimated) difference in the crash probabilities due to the degeneracies from the stabilizer elements.

In each of the cases, the noise at threshold for the actual non-local noise p_a is found to have the same e_2 as the p_e noise, with the same noise on each qubit. It is interesting to note that for all of these cases, the entropy at the 2nd level of the code, e_2 , is the same; therefore to determine the threshold for a particular type of noise for the $[[7, 1, 3]]$, one just has to find when the entropy of the logical noise at the second level of the code is $e_2 = 1.0011(8)$.

So far we have only considered the non-local effects of post-selection from the first level. What about the

TABLE IX: Values at the hashing bound. Refer to Tab. IV for descriptions of the types of noise.

Value	Depolarizing	Knill	Forward
p	8.27515%	6.90240%	4.81816%
p_s	5.65%	4.80%	6.18%
p_d	9.56%	9.67%	10.40%
p_g	0.63%	0.54%	0.76%

second level? In addition to the stabilizers in each of the sub-blocks from the previous level of the code, there are new stabilizers that consist of an encoded \overline{X} in 4 of the 7 sub-blocks. The lowest weight for these is 12, which consist of distance and weight 3 errors in each of 4 sub-blocks. There are 7 of these stabilizers at the top level of the code, so there are 7 ways to choose the sub-blocks. Each of the sub-blocks has 7 elements of distance and weight 3. Therefore there are 7^5 stabilizers of distance 12. The estimated difference in the crash probability is $c_e = 3 \times 7^5 \binom{12}{6} p_g^6$.

For the forward type noise $c_e = 6.5 \times 10^{-6}$, which results in a difference in p_f of less than 10^{-6} , so the final actual threshold for the forward type noise for the $[[7, 1, 3]]$ code is $p_f = 4.6699\%$. The effect on the other types of noise is insignificant.

D. The $[[23, 1, 7]]$ CSS code

The $[[23, 1, 7]]$ CSS code has 506 weight 8 elements corresponding to no error [20], i.e., 506 weight 8 stabilizer elements. Using the method of the previous section, the estimated difference in crash probability is

$$c_e = 3 \binom{8}{4} 506 p_g^4.$$

For the forward type noise near the hashing bound, this results in $c_e = 0.00035$.

Again, using the noise notation of Appendix C, the undetected distance 7 elements have weights 7, 11, 15, and 23 [20], so the noise map for a logical crash is of the form $f_{23}(x) = c_7 x^7 + c_{11} x^{11} + c_{15} x^{15} + c_{23} x^{23}$. Since no error input results in no error output, $f(1) = 1$. Since this code can correct any distance 1, 2, or 3 error, it follows that $f'(1) = f''(1) = f'''(1) = 0$. Solving these gives the channel map of

$$f_{23}(x) = \frac{3795}{512} x^7 - \frac{805}{64} x^{11} + \frac{1771}{256} x^{15} - \frac{385}{512} x^{23}.$$

The crash probability is $\frac{1-f_{23}(x)}{2}$. For the same noise on each qubit, we found the threshold to be $p_e = 4.805(4)\%$, and $x = 0.78064$. Solving for when the difference in crash probability is $\Delta = 0.00035$, we get $p_r = 4.801(4)\%$, only lowering the threshold by 4×10^{-5} in p_f . The actual threshold was calculated to be $p_a = 4.800(3)\%$.

TABLE X: $[[23, 1, 7]]$ thresholds. Refer to Tab. IV for descriptions of the types of noise.

Value	Depolarizing	Knill	Forward
p_e	8.25(3)%	6.88(2)%	4.805(4)%
c_e	0.00017	0.00009	0.00035
Δ_p	0.003%	0.002%	0.0040%
p_a	-	-	4.800(3)%
p_r	8.25(0)%	6.88(0)%	4.801(4)%

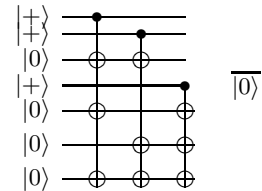
These calculations were also performed for Knill and depolarizing noise in Tab. X. For the Forward type noise, $p_e = 4.805(4)\%$, which results in a total noise of $p = 10.968\%$, the threshold for one type of noise for the $[[23, 1, 7]]$ code. The logical entropy after the first level is $e_1 = 1.0016\%$. For the other types of noise, the values of p_e were calculated by determining which noise gave this same logical entropy of e_1 after the first level. Based upon calculations with change in entropy near the threshold under different levels of concatenation for the $[[7, 1, 3]]$ and $[[17, 1, 5]]$ CSS codes, this method should be accurate to at most a couple times 10^{-5} for the entropy and a small fraction of 10^{-5} for the thresholds.

E. Encoding and post-selection process

p The methods described earlier in this paper rely on the code being concatenated many times with itself. A possible problem occurs if we create encoded $\overline{|0\rangle}$ for the $[[7, 1, 3]]$ code like in Fig. 9, and post-select as described in this paper, then the Z noise will converge to the completely depolarizing case, where each of the 8 error syndromes are equally likely. The reason for this problem is that \overline{Z} is now a stabilizer, and there are now 7 stabilizer elements of weight 3.

To correct this problem, instead of post-selecting just one logical qubit after the first level of the $[[7, 1, 3]]$ code, we first create a logical $\overline{|0\rangle}$ state for the $[[7, 1, 3]]$ code as in Fig. 9 and again in the Hadamard basis, so that we have the logical states $\overline{|0\rangle}$ and $\overline{|+\rangle}$. We apply a logical

FIG. 9: Creating encoded $\overline{|0\rangle}$ for the $[[7, 1, 3]]$ CSS code. This results in the state $\overline{|0\rangle} = \frac{1}{\sqrt{8}}(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle)$.



CNOT transversally from each of the physical qubits of the $|+\rangle$ state to the $|0\rangle$ state to create an encoded Bell pair $\frac{1}{\sqrt{2}}|00\rangle + |11\rangle$ that uses 14 physical qubits. Now there are no longer any stabilizer elements of weight 3. The 7 weight 4 ones were discussed in Sec. VB. The next lowest weight stabilizers are the 49 weight 6 stabilizers that correspond to $\bar{Z} \otimes \bar{Z}$.

Post-selection can now be used to refine this state. To post-select, we start with two Bell pair states. We apply logical CNOT from the first logical qubit of the first Bell pair to the second logical qubit of the second Bell pair. Similarly, we apply logical CNOT from the second logical qubit of the first Bell pair to the first logical qubit of the second Bell pair. When applying the logical CNOT gates, we mix up the orders of the qubits, to even out the errors, which is better for post-selection. There are 168 ways to permute the 7 qubits of the $[[7, 1, 3]]$ CSS code and preserve the code. Let $p(x)$ represent such a permutation. The distance 3 codewords are those qubits q_1, q_2 and q_3 such that their parity is even, that is, $q_1 \oplus q_2 \oplus q_3 = 0$. The parity of $p(q_1), p(q_2)$ and $p(q_3)$ is also even, and $p(1), p(2)$ and $p(4)$ determine the permutation.

We then measure the second Bell pair in the standard basis, rejecting the first Bell pair if an error is detected. We apply the Hadamard gate transversally to the remaining (first) Bell pair, to exchange bit flip errors and phase flip errors for the next round of post-selection. Notationally, we consider the initial encoding of the Bell pair to be the 0 times post-selected states. To create the n times post-selected state, we use an $n - 1$ times post-selected state as the source (1st) Bell pair, and a d_n times post-selected state for the destination (2nd) Bell pair. $d_n = 0, 0, 1, 2, 3, \dots$ is a well-performing sequence for post-selection.

After a sufficient amount of post-selection is performed, we take 7 Bell pairs, and measure one of the logical qubits in the Hadamard basis on the 2nd qubit, rejecting if there is an error. We now have 7 post-selected encoded qubits. We use these qubits to create the $|0\rangle$ state for the $[[7, 1, 3]]$ code encoded with itself (using a total of 49 qubits), and use more Bell pairs to detect if there are any errors after each step, rejecting everything if an error is detected. We then construct an encoded Bell pair, using 98 physical qubits, and repeat the process described above for post-selection, rejecting everything if there is an error detected in a single qubit.

Once we have done this for a number of levels and the rate of errors is sufficiently low, we have two options. One is that we continue to use the $[[7, 1, 3]]$ code, and we construct the more complicated encoded ancilla shown in Fig. 7. The other possibility is that we prefer to use some other code, such as the $[[23, 1, 7]]$ CSS code. In this case, we now encode into that code. However, we wish to remove the levels of the $[[7, 1, 3]]$ CSS code from the encoding, since the $[[7, 1, 3]]$ code might not have as good a threshold (when we actually use error correction, not this post-selection process). We can decode one level at a time, using the following process for decoding the

$[[7, 1, 3]]$ CSS code. Suppose we have the encoded state $c_0|0\rangle + c_1|1\rangle$. We can represent this as

$$\begin{aligned} m_1 &= x_1 \oplus e & m_2 &= x_2 \oplus e \\ m_3 &= x_2 \oplus x_1 \oplus e & m_4 &= x_4 \oplus e \\ m_5 &= x_4 \oplus x_1 \oplus e & m_6 &= x_4 \oplus x_2 \oplus e \\ m_7 &= x_4 \oplus x_2 \oplus x_1 \oplus e, \end{aligned}$$

where x_1, x_2, x_4 are summed over 0 and 1, and e represents the encoded basis states. We post-select it for Z errors. Now we measure the last 4 qubits, rejecting everything and starting over if the parity isn't even. $m_4 \oplus m_5 = x_1$ and $m_4 \oplus m_6 = x_2$, so by applying X gates to the first 3 qubits conditional on these measurements, we have $m_1 = m_2 = m_3 = e$, which is $c_0|000\rangle + c_1|111\rangle$. We then measure two of the qubits in the Hadamard basis. To avoid introducing more error, we can reject if either one is in the $|-\rangle$ state. For the forward type noise near the hashing bound, the total bit flip error is about $3p_g \approx 2\%$, and the total phase flip error is about $p_b \approx 6\%$. Once we have removed the lowest level of the $[[7, 1, 3]]$ code, we run post-selection repeatedly, and then remove another level of the $[[7, 1, 3]]$ CSS code. Eventually, we remove all of the levels of the $[[7, 1, 3]]$ code and are left with a logical ancilla state in our desired code, which may be the $[[23, 1, 7]]$ code concatenated with itself repeatedly.

VI. DISCUSSION OF OVERHEAD

In Sec. V, we found that the degeneracies of a code cause the errors on each qubit to cease being independent of each other, thereby lowering the threshold a bit. In this section, we will discuss how using finite resources affects the thresholds. We pick some noise rate below the threshold and a rate of error that we are willing to tolerate per logical gate, which results in a finite amount of overhead involved. We examine how the noise converges below the threshold.

A. Post-selection overhead

Suppose that there are q logical qubits encoded in an n qubit code, giving a total of $N = qn$ qubits. If the total bit flip noise on the destination qubits before measurement is p , then the probability of not rejecting everything is roughly $p_k \approx (1 - p)^N$, and so we have to run this an average of $\frac{1}{p_k}$ times. The process described in Sec. VE results in an overhead of $o_s = \frac{1}{p_k}(o_{t-1} + o_{t-2})$ after s post-selections. Then, roughly o_s is $o(N(p_k^{-1} + 1 - p_k)^s)$. After a while, the noise will tend toward an equilibrium point, getting a factor r closer each time. Therefore, to get within ϵ of the equilibrium point requires s to be of the order $\log_r \epsilon$. If N is large, to get within

TABLE XI: Fidelity the same at 0th and 1st levels of code. Refer to Tab. IV for descriptions of the types of noise.

Code	Noise	p	fidelity
[[7, 1, 3]]	Knill	3.472%	0.90602
[[7, 1, 3]]	Depolarizing	4.039%	0.91122
[[7, 1, 3]]	Forward	2.9595%	0.87703
[[23, 1, 7]]	Forward	3.5471%	0.85108

ϵ of the equilibrium point will require something like $N(p_k^{-1})^{\log_r \epsilon} = N\epsilon^{-N \log_r(1-p)}$ overhead.

At the hashing bound for forward type noise, $p \approx 15.3$, so for Bell pairs for the $[[7, 1, 3]]$ code, the predicted probability of not rejecting is $p_k \approx (1-p)^{14} = 9.79\%$. The actual $p_k = 9.21\%$. The observed rate of convergence is $r = 0.766$. This results in $o(N\epsilon^{-0.665N})$ overhead. This is exponential overhead in the number of qubits, and doubly exponential in the number of levels of concatenation of the code.

B. Convergence by level

Suppose we have a distance d code concatenated with itself l times, the initial entropy was t , and the threshold entropy was t_c . Near the threshold, the entropy behaves roughly as $t \rightarrow t_c - (t_c - t)(d^l)^\alpha$, where $\alpha = \log_2 \log_2 e \approx 0.53$.

Away from the threshold, if $\Delta = \frac{(t_c - t)d^j \alpha}{t_c}$, the rate of logical error has a magnitude of around 10^{-3} or 10^{-4} for $\Delta = 2$, and goes to 0 very rapidly after this.

For example, suppose we have the $[[7, 1, 3]]$ CSS code concatenated with itself $l = 4$ times and our noise is of the form $(0, 0, p)$, and p is $\frac{2}{3}$ of the threshold value of 10.963(2)%. Then $d = 3$, $t_c = 0.49880$, and $t = 0.3492$. Therefore, $\Delta \approx 2.5$. The probability of an encoded error is 2×10^{-5} .

If the noise is some ϵ below the threshold, then to get the noise below a given value requires that d^l be $o(\epsilon^{-\alpha^{-1}})$. If the code encodes 1 qubit into n qubits, then the number of qubits n^l is $o(\epsilon^{-\frac{\log_d n}{\alpha}})$. For the $[[7, 1, 3]]$ code, this is $o(\epsilon^{3.35})$.

C. Fixed fidelity points

Note that just below the threshold, the rate of an error increases for the first few levels of concatenation. Suppose we require that the rate of error be the same after the first level of the code as it was from the noise in Eq. 6. For the Knill type noise with the $[[7, 1, 3]]$ CSS code, $p = 3.472\%$ has a fidelity of $p_I = 0.90602$ in both cases. This can be thought of as a very crude proof that $p = 3.472\%$ is not above the threshold. Other values are calculated in Tab. XI

VII. CONCLUSION

We find that for various CSS codes, particularly the $[[23, 1, 7]]$ CSS code, when extremely large amounts of resources (i.e. remotely interacting ancilla qubits) are available, the threshold is close to that of the hashing bound (when the Shannon entropy of the noise is 1). We conjecture that the hashing bounds given in Table V represent an upper bound for fault tolerant thresholds. This bound is much lower than the best previously known upper bounds of 45% [3] and 29.3% [13]. It is higher than a fault tolerant threshold for the adversarial noise model [12].

The CSS codes are formed from classical codes and therefore do not exceed the Shannon hashing rating bound. The hashing bound had been conjectured to be the upper bound for non fault tolerant error correction, but it was found that codes can exceed this [5, 9, 22, 23]. However, the codes that exceed the hashing bound seem to do so because of their degeneracies, and these degeneracies substantially hurt the threshold, as is seen in Sec. V. The bit flip codes, which seem to yield the highest capacities [9], have $\binom{n}{2}$ ways of having Z errors on two qubits, which results in no error. This effect on the threshold is large enough that the threshold will be below the hashing bound.

We conjecture that if CNOT is the only multiple qubit gate implemented directly, the fault tolerant hashing bound described in this paper serves as an upper bound on the fault tolerant threshold. As can be seen from Tab. X, the $[[23, 1, 7]]$ CSS code provides high fault tolerant thresholds, which are at least 99.6% of the hashing bound thresholds given in Tab. V, leaving little room for improvement.

In the future, we hope to apply these ideas to analyze the amount of overhead, and develop new realistic fault tolerant schemes. Unfortunately, it is unrealistic to get close to the fault tolerant thresholds given here. The resources needed to run quantum computing very close to these thresholds will be exceedingly high; to do so would require googols of universes with remotely interacting qubits. However, it would be useful to compare schemes with restricted amounts of overhead to the truly ultimate thresholds found in this paper. Since the thresholds found here are so high, achieving even some fraction of them with a reasonable amount of overhead may offer an improvement over current schemes. For example, we find a threshold of 6.9% for Knill noise, which is much larger than the 1% threshold with a reasonable amount of overhead found by Knill [16].

Acknowledgments

We thank K. B. Whaley for helpful discussions, and careful reading of the manuscript. Quantum circuits were drawn using Q-Circuit [6].

-
- [1] C. Bennett, G. Brassard, C. Crepau, R. Jozsa, A. Pereg, and W. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys Rev. Lett.*, 70:1895, 1993.
 - [2] Sergei Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, 2005. quant-ph/0403025.
 - [3] H. Buhrman, R. Cleve, M. Laurent, N. Linden, A. Schrijver, and Falk Unger. New limits on fault-tolerant quantum computation. In *Conf. on Foundations of C.S. (FCS'06)*, page 411, 2006. quant-ph/0604141.
 - [4] I. L. Chuang and M. A. Nielsen. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.
 - [5] D. P. DiVincenzo, P. W. Shor, and J. Smolin. Quantum channel capacity of very noisy channels. *Phys. Rev. A*, 57(2):830, 1998. quant-ph/9706061.
 - [6] B. Eastin and S. T. Flammia. Q-circuit tutorial. quant-ph/0406003.
 - [7] J. Fern. Correctable noise of quantum error correcting codes under adaptive concatenation. *Phys. Rev. A*, 77:010301, 2008. quant-ph/0703258.
 - [8] J. Fern, J. Kempe, S. Simić, and S. Sastry. Generalized performance of concatenated quantum codes – a dynamical systems approach. *IEEE Trans. Autom. Control*, 51:448, March 2006. quant-ph/0409084.
 - [9] J. Fern and K. B. Whaley. New lower bounds on the non-zero capacity of pauli channels, 2007. arXiv:0708.1597.
 - [10] D. Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57:127–137, 1998. quant-ph/9702029.
 - [11] M. Grassl. Lower bounds and encoding circuits for weakly self-dual css codes. <http://avalon.ira.uka.de/home/grassl/QECC/CSS>.
 - [12] A. Kay. Bounding fault-tolerant thresholds for purification and quantum computation. arXiv:0705.4360, 2007.
 - [13] J. Kempe, O. Regev, F. Unger, and R. de Wolf. Upper bounds on the noise threshold for fault-tolerant quantum computing. arXiv:0802.1464, 2008.
 - [14] E. Knill. Fault-tolerant postselected quantum computation: Schemes. quant-ph/0402171.
 - [15] E. Knill. Quantum computing with realistically noisy devices. quant-ph/0410199.
 - [16] E. Knill. Quantum computing with realistically noisy devices. *Nature*, 434:39–44, 2005.
 - [17] J. Preskill. Fault tolerant quantum computation. quant-ph/9712048, 1997.
 - [18] B. Rahn, A. C. Doherty, and H. Mabuchi. Exact performance of concatenated quantum codes. *Phys. Rev. A*, 66(032304), 2002. quant-ph/0206061.
 - [19] B. W. Reichardt. *Error-detection-based quantum fault tolerance against discrete Pauli noise*. PhD thesis, UC Berkeley, 2006. quant-ph/0612004.
 - [20] Ben W Reichardt. Improved magic states distillation for quantum universality. *Quantum Inf. Proc.*, 4:251, 2005. quant-ph/0411036.
 - [21] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379, 1948.
 - [22] P. W. Shor and J. Smolin. Quantum error-correcting codes need not completely reveal the error syndrome. quant-ph/9604006.
 - [23] G. Smith and J. Smolin. Degenerate quantum codes

for pauli channels. *Phys. Rev. Lett.*, 98(3):030501, 2007. quant-ph/0604107.

APPENDIX A: CODE METHOD

In this section, we use the quantum error correction channel formalism of [8] to find the results of most of Sec. II by an alternate means.

Suppose we have a code that encodes k qubits into n qubits. We have an operator \mathcal{E} which encodes 4^k dimensional density vectors representing density states on k qubits into 4^n dimensional density vectors (as shown in Eq. 1) representing density states on n qubits. The noise that occurs on these n qubits is represented by a 4^n by 4^n dimensional operator \mathcal{N} . This is followed by measuring the stabilizer operators to determine the syndrome i . We next apply a recovery operator R_i and decode to evaluate the resulting logical errors. This is the channel map technique described in [7, 8, 18].

The resulting noise map for each recovery operator R_i is

$$\mathcal{G}^{R_i} = \frac{1}{2^{n-k}} \mathcal{E}^\dagger \circ \mathcal{O}(R_i) \circ \mathcal{N} \circ \mathcal{E}.$$

The total noise map is $\mathcal{G} = \sum_i \mathcal{G}^{R_i}$.

Now we use these channel maps for a different way of looking at Sec. II. If the physical data qubit was $\alpha|0\rangle + \beta|1\rangle$, we model this by having the 2 physical qubits in the state $\alpha|00\rangle + \beta|11\rangle$, apply CNOT, and then measure. 0 represents no error and 1 represents an error, just like in Fig. 3. If this process is noise free, the data qubit is now $\alpha|0\rangle + \beta|1\rangle$.

In Fig. 3, we assumed that the source qubit has a pre-existing noise \mathcal{S} , and the destination qubit has a pre-existing noise \mathcal{D} . We then apply CNOT, and the noise \mathcal{Q} is created. We follow this by measuring the destination qubit.

We look at the two codes in Tab. XII. C1 is the two qubit bit flip code. By applying CNOT from its first qubit to its second qubit, we change it into the C2 code, and vice versa.

These can be written out in matrix form. For example for C2,

$$\mathcal{E}^\dagger = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Let $\mathcal{O}(\text{CNOT})$ be the superoperator for CNOT. Together with its noise operator, the faulty CNOT gate is $\mathcal{Q} \circ \mathcal{O}(\text{CNOT})$. Together with the pre-existing noise, we have

$$\mathcal{Q} \circ \mathcal{O}(\text{CNOT}) \circ \mathcal{S} \otimes \mathcal{D}. \quad (\text{A1})$$

TABLE XII: Description of two codes

	C1	C2
$ 0\rangle$	$ 00\rangle$	$ 00\rangle$
$ 1\rangle$	$ 11\rangle$	$ 10\rangle$
S	$\{II, ZZ\}$	$\{II, IZ\}$
\mathcal{E}_I	$II + ZZ$	$II + IZ$
\mathcal{E}_X	$XX + YY$	$XI + XZ$
\mathcal{E}_Y	$XY + YX$	$YI + YZ$
\mathcal{E}_Z	$ZI + IZ$	$ZI + ZZ$
R	$\{II, IX\}$	$\{II, IX\}$

Note that this changes from code C1 to code C2. If we wish to do the calculations in terms of code C1, we apply **CNOT** at the end. If we wish to do the calculations in terms of code C2, we apply **CNOT** at the beginning. The logical noise from these codes are

$$\mathcal{N}_{C1} = \mathcal{O}(\text{CNOT}) \circ \mathcal{Q} \circ \mathcal{O}(\text{CNOT}) \circ \mathcal{S} \otimes \mathcal{D}$$

$$\mathcal{N}_{C2} = \mathcal{Q} \circ \mathcal{O}(\text{CNOT}) \circ \mathcal{S} \otimes \mathcal{D} \circ \mathcal{O}(\text{CNOT}).$$

By conjugating, **CNOT** produces the following map c which acts on two qubit Pauli matrices as $c(\sigma)$ as

$$(II), (IX), IY \leftrightarrow ZY, IZ \leftrightarrow ZZ, XI \leftrightarrow XX$$

$$XY \leftrightarrow YZ, XZ \leftrightarrow YY, YI \leftrightarrow YX, (ZI), (ZX).$$

This map has order 2, that is, $c(c(\sigma)) = \sigma$. Note that bit flips (X or Y) propagate forward as X , and phase flips (Y or Z) propagate backwards as Z .

1. Diagonal noise

We now use the shorthand $\mathcal{N}_\sigma = \mathcal{N}_{\sigma,\sigma}$. The initial source noise has diagonal noise $\mathcal{S} = [1, S_X, S_Y, S_Z]$, and similarly the destination noise has the diagonal noise $\mathcal{D} = [1, D_X, D_Y, D_Z]$. Control can be used to make the noise diagonal, as was done with the Singular Value Decomposition Theorem for channels [8]. This turns one qubit unital noise into diagonal noise, and any non-unital part has little effect on the threshold.

If there is no error detected, and σ and σ' are one qubit Pauli matrices, then

$$\mathcal{G}_{\sigma,\sigma'}^{II} = \frac{1}{2}(N_{\sigma I, \sigma' I} + N_{\sigma I, \sigma' Z} + N_{\sigma Z, \sigma' I} + N_{\sigma Z, \sigma' Z})$$

$$\mathcal{G}_{\sigma,\sigma'}^{XI} = \frac{1}{2}(N_{\sigma I, \sigma' I} + N_{\sigma I, \sigma' Z} - N_{\sigma Z, \sigma' I} - N_{\sigma Z, \sigma' Z}).$$

In the case of diagonal noise, we have Eq. 5.

APPENDIX B: OTHER GATES AND FAULT TOLERANT TRANSVERSAL GATES

For the doubly even (stabilizers have weights divisible by 4) CSS codes discussed in this paper, the Clifford gates

are encoded as themselves (or their adjoint) transversally [10]. Measurement of a Pauli operator and **CNOT** are also encoded as themselves transversally. Together with any 1 qubit pure non Pauli eigenstate, these give universal quantum computation [20].

Transversely encoded gates are very useful to prevent propagation of errors. For a more in depth discussion of how these are useful for fault tolerance, see [17].

Typically, one prepares a "magic" state, $\frac{1}{\sqrt{2}}(|0\rangle + \sqrt{i}|1\rangle)$ or $\cos(\frac{\pi}{8})|0\rangle + \sin(\frac{\pi}{8})|1\rangle$ that can be used to create the $\frac{\pi}{8}$ gate. We discuss 3 approaches to get a "magic state":

- The method described by Nielsen and Chuang to measure the state using a cat state [4].
- There is a 15 qubit Hamming CSS code, where the $\frac{\pi}{8}$ gate can essentially be encoded as itself [2]. If there is a Bell pair consisting of the 15 qubit CSS code and some other code, this can be used to teleport the logical qubit into another code. A similar method would be noting that the 15 qubit code is similar to the $[[7, 1, 3]]$ CSS code, and converting it directly into that code.
- We have a high distance code encoded in a Bell pair. One of the logical qubits is decoded into a physical qubit [14], and then the $\frac{\pi}{8}$ gate is applied to that qubit, followed by a measurement in the X eigenbasis. Then, the code is encoded into the 15 qubit code, and post-selection is run on that as in [2].

APPENDIX C: NOISE

In this section, we discuss our representation of noise, and derive a formula for the noise after post-selection.

If there is probability p_x of an error, we write the noise as $x = 1 - 2p_x$.

Since $x = (1 - 2p_y)(1 - 2p_z) = 1 - 2(p_y + p_z - 2p_y p_z)$, two noises y and z are combined as $x = yz$.

Suppose a qubit with noise x_1 has **CNOT** applied to a qubit with noise x_2 , and post-selection is performed by measuring the 2nd qubit in the standard basis. We either have no error with probability $p_g = (1 - p_1)(1 - p_2)$, or an undetected error with probability $p_b = p_1 p_2$. The new probability of an error is then $p = \frac{p_b}{p_g + p_b}$, and so

$$\text{post } (x_1, x_2) = 1 - 2p = \frac{p_g - p_b}{p_g + p_b}$$

$$= \frac{2 - 2p_1 - 2p_2}{2 - 2p_1 - 2p_2 + 4p_1 p_2} = \frac{x_1 + x_2}{1 + x_1 x_2}.$$

APPENDIX D: $[[7, 1, 3]]$ CSS CODE

In this section, we find non fault tolerant probabilities of errors for the $[[7, 1, 3]]$ CSS code, which are applied

TABLE XIII: Distances d of errors of weight w (errors on w qubits) for the $[[7, 4, 3]]$ classical code that generates the $[[7, 1, 3]]$ CSS code.

w	$d = 0$	$d = 1$	$d = 2$	$d = 3$	Tot. cases
0	1	0	0	0	1
1	0	7	0	0	7
2	0	0	21	0	21
3	0	28	0	7	35
4	7	0	28	0	35
5	0	21	0	0	21
6	0	0	7	0	7
7	0	0	0	1	1

earlier for fault tolerant calculations.

The $[[7, 1, 3]]$ CSS code is generated from the classical $[[7, 4, 3]]$ Hamming code given by the parity check matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \quad (\text{D1})$$

The stabilizer generators for the $[[7, 1, 3]]$ CSS code are

$$\begin{aligned} g_1 &= IIIXXXX & g_4 &= IIIZZZZ \\ g_2 &= IXXIIXX & g_5 &= IZZIIZZ \\ g_3 &= XIXIXIX & g_6 &= ZIZIZIZ \end{aligned}$$

The encoded Pauli operators are

$$\overline{X} = X^{\otimes 7}, \overline{Z} = Z^{\otimes 7}, \overline{Y} = i\overline{X}\overline{Z} = -Y^{\otimes 7}$$

A circuit for generating the encoded $|\overline{0}\rangle$ is given in Fig. 9.

1. Probabilities of errors of a given distance

Suppose we have the $[[7, 1, 3]]$ CSS code, with uncorrelated bit flip and phase flip noise. Then the two types of noise can be corrected independently of each other. If we wish to find the probability m_i of a distance i phase flip or bit flip error, we need just consider the $[[7, 4, 3]]$ classical code that generates the CSS code.

Looking at the parity check in Eq. D1, we see that there are 8 different possible error syndromes. The 0 syndrome corresponds a distance 0 error (no error) or distance 3 error (undetected error). The other syndromes correspond to either a distance 1 or distance 2 error. The code corrects any one error. Counting the 2^7 different possible errors, and their distances, we get Tab. XIII.

In terms of the quantum stabilizer code, distance 0 errors (no error) correspond to members of the stabilizer group S , which preserve the codespace. Distance 3 errors are encoded errors that commute with the stabilizer elements, and so are in $C(S) \setminus S$, where the C represents

centralizer. Distance 1 and 2 errors are errors outside of the codespace.

If we have independent probability p of an error on each qubit, then the probability of a distance d error is m_d , where

$$\begin{aligned} m_0 &= (1-p)^7 + 7p^4(1-p)^3 \\ m_1 &= 7p(1-p)^6 + 28p^3(1-p)^4 + 21p^5(1-p)^2 \\ m_2 &= 21p^2(1-p)^5 + 28p^4(1-p)^3 + 7p^6(1-p) \\ m_3 &= 7p^3(1-p)^4 + p^7. \end{aligned}$$

If we use $x = 1 - 2p$ instead, we get

$$\begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 7 & 7 & 1 \\ 7 & 7 & -7 & -7 \\ 7 & -7 & -7 & 7 \\ 1 & -7 & 7 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x^3 \\ x^4 \\ x^7 \end{bmatrix}. \quad (\text{D2})$$

Alternately, we see from the centralizer that $m_0 = \frac{1+7x^3+7x^4+x^7}{16}$. This represents a distance 0 error (no error). What happens to each of the terms as we add an error of a fixed d ? We find all the ways to add an error, and replace x with $-x$ on that particular qubit, and then average the result. We get

$$\begin{array}{c|c|c|c} d=0 & d=1 & d=2 & d=3 \\ \hline 1 & 1 & 1 & 1 \\ x^3 & \frac{1}{7}x^3 & -\frac{1}{7}x^3 & -x^3 \\ x^4 & -\frac{1}{7}x^4 & -\frac{1}{7}x^4 & x^4 \\ x^7 & -x^7 & x^7 & -x^7 \end{array}. \quad (\text{D3})$$

For this code, an encoded error is a distance 2 or 3 error. Since the 0 syndrome is either distance 0 or 3, and the other 7 error syndromes are each either distance 1 or distance 2, and the syndrome calculation is linear over F_2^7 , we get 16 equivalence classes of 8 errors each. We can choose a representative element of each, for example all of the 0, 1, 6, and 7 qubit Z errors. We then calculate what happens when errors are combined by finding the distance of the combined error. For example, if we pick 2 random distance 1 errors, we have a $\frac{1}{7}$ probability of a distance 0 error, and a $\frac{1}{6}$ probability of a distance 2 error. Then, if we combine a probability a_d of having distance d errors with a probability b_d of having distance d errors, we get a combined probability a'_d of having distance d errors

$$\begin{bmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} = \begin{bmatrix} b_0 & \frac{1}{7}b_1 & \frac{1}{7}b_2 & b_3 \\ b_1 & b_0 + \frac{6}{7}b_2 & b_3 + \frac{6}{7}b_1 & b_2 \\ b_2 & b_3 + \frac{6}{7}b_1 & b_0 + \frac{6}{7}b_2 & b_1 \\ b_3 & \frac{1}{7}b_2 & \frac{1}{7}b_1 & b_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}. \quad (\text{D4})$$

If we post-select on a and b , then we keep the result with probability

$$p_k = a_0b_0 + \frac{1}{7}a_1b_1 + \frac{1}{7}a_2b_2 + a_3b_3,$$

and the result is

$$\frac{1}{p_k} \begin{bmatrix} a_0 b_0 \\ \frac{1}{7} a_1 b_1 \\ \frac{1}{7} a_2 b_2 \\ a_3 b_3 \end{bmatrix}. \quad (\text{D5})$$